

Linguagem de Programação II

Profº Jânio Eduardo

janio.Eduardo@fbr.edu.br

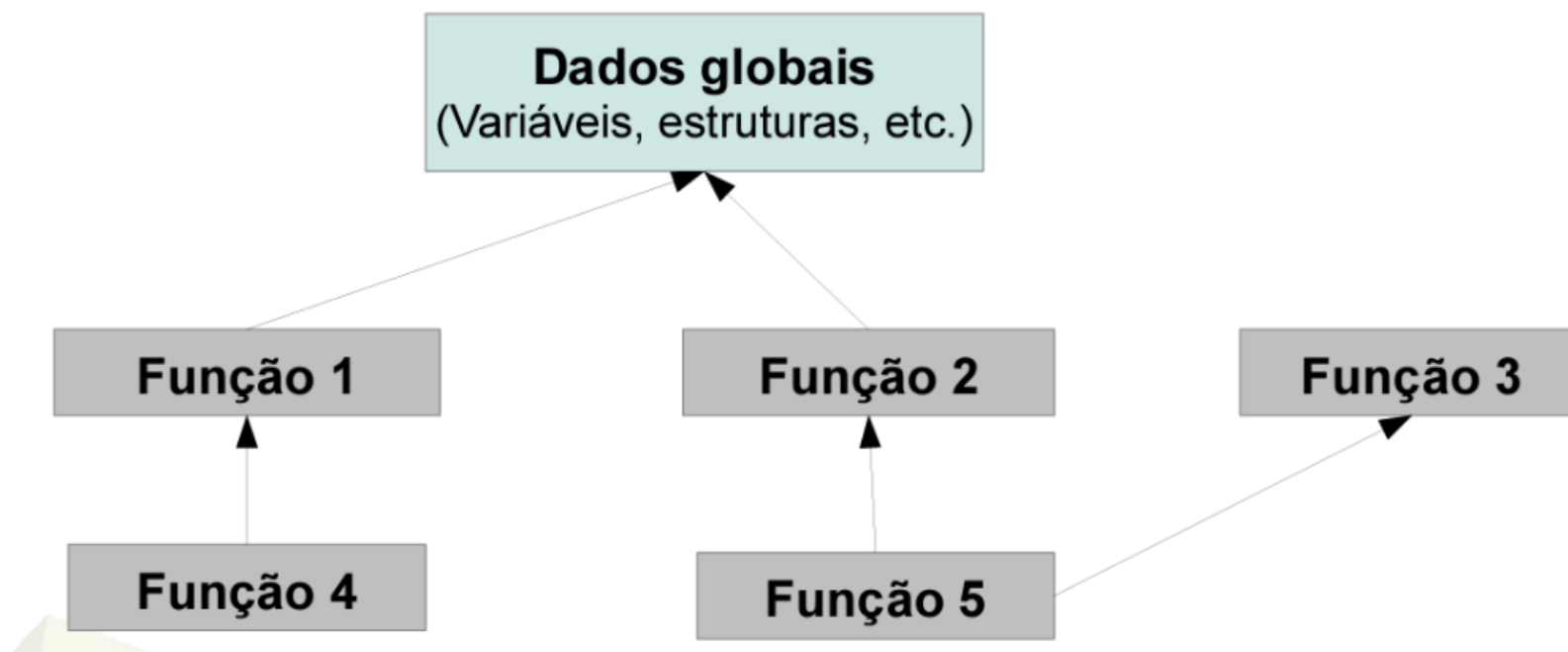
Roteiro de hoje

- ▶ Revisão da programação Estruturada
- ▶ Compreensão dos conceitos iniciais;

Programação Estruturada

- ▶ Modelo de programação que utilizamos até o momento
- ▶ Em programação imperativa temos:
 - Dados do programa (variáveis)
 - Ações que operam sobre esses dados (funções)
- ▶ Normalmente as funções são reunidas em uma ou mais bibliotecas

Programação estruturada



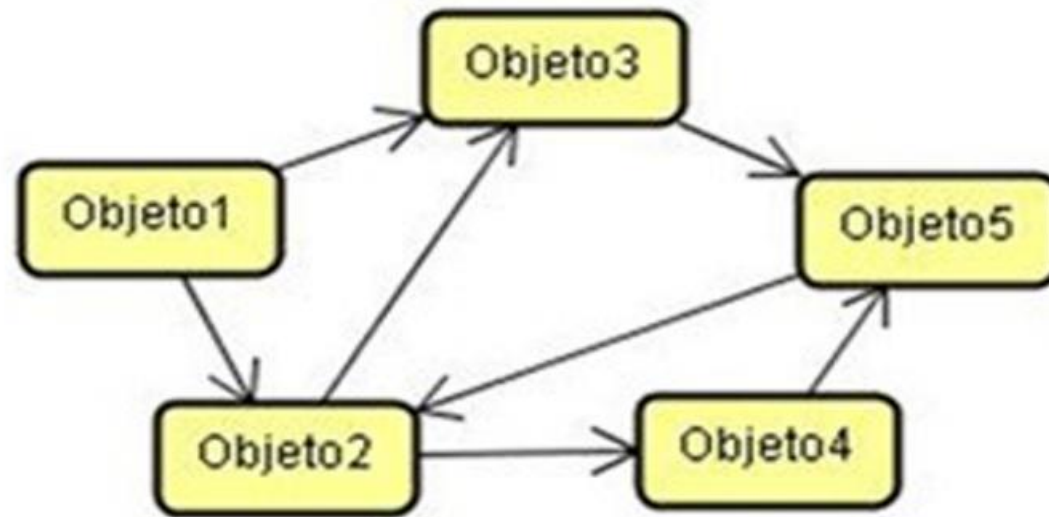
Características da programação estruturada

- ▶ Dados e funções são independentes;
- ▶ Dados são fornecidos a funções, que normalmente retornam um resultado;
- ▶ Há pouco (ou nenhum) controle ao acesso a variáveis e funções
- ▶ Ainda hoje esse modelo de programação é utilizado
- ▶ É também base para modelo de orientação à objetos

Programação orientada a objetos

- ▶ Forma de pensar em um programa a partir de abstrações da vida real:
 - Objetos;
 - Troca de mensagens
 - Composição;
 - Hierarquia, etc.
- ▶ O Programa é composto por um conjunto de objetos que se comunicam entre si através da troca de mensagens entre eles

POO



- ▶ Dados e funções:
 - São definidos e encontram-se interligados dentro de um objeto ;
 - Podem (e devem) ser protegidos de acesso indevido (conceito de encapsulamento)

POO – Princípios básicos

- ▶ Abstração
- ▶ Classes e Objetos
 - São as entidades básicas de um sistema OO
 - Objetos são criados a partir de classes
- ▶ Outros conceitos:
 - Encapsulamento
 - Composição
 - Herança
 - Polimorfismo



Abstração

- ▶ Consiste em uma forma de pensar na solução de um problema de modo a gerenciar a sua complexidade
- ▶ A partir da descrição de um problema, deve-se focar e “modelar” apenas os aspectos essenciais
- ▶ Exemplos no mundo real:
 - Um médico especialista em uma determinada área
 - Um engenheiro que projeta um edifício pensando em cada parte isoladamente (sistema elétrico, hidráulico, etc...).

Objetos

- ▶ Representam objetos do mundo real ou computacional
 - Aluno, venda, biblioteca, calculadora, carro, veículo, etc.
- ▶ Conjunto de elementos que fazem parte da solução que estamos desenvolvendo
 - Loja de veículos: objetos vendedor, carro, moto, venda, etc.
 - Biblioteca: livro, exemplar, empréstimo, funcionário, etc.
- ▶ Possuem características e comportamento.

Objetos

- ▶ Característica (atributos, em OO)
 - Ex.: Atributos de “Pessoa”: nome, cpf, identidade, idade, estado civil...
 - Atributos de “Carro”: cor, motor, ano, marca, etc.

- ▶ Comportamento (métodos, em OO)
 - Ações que podem ser realizadas pelo objeto;
 - Comportamentos de um “Carro”: ligar, acelerar, diminuir a velocidade, parar, etc.

Exercícios

- ▶ Escolha um objeto a sua volta e cite alguns de seus atributos e comportamentos.

Classes

- ▶ Pode ser vista como um molde ou projeto para se criar “objetos” daquela classe
- ▶ Os objetos são representações concretas (instâncias) das classes
- ▶ Uma classe criada *define um novo* “tipo” de dados.

“Classes” no mundo real

- ▶ No mundo real temos vários exemplos de projetos, por exemplo:
 - Uma planta de engenharia,
 - Um desenho,
 - Uma maquete de um prédio, etc.
- ▶ No entanto, uma classe pode ser a representação para criar qualquer tipo de objetos
 - Uma pessoa, um carro, uma caneta, um livro, uma loja, uma nota fiscal, um item de compra, etc.

Exemplos



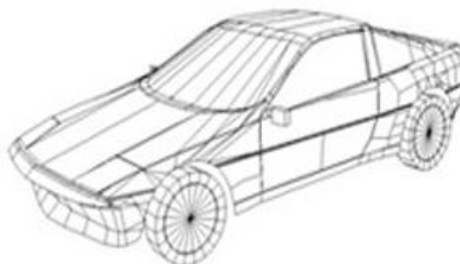
Exemplos



Outra representação de uma casa

Classe carro e objetos do tipo carro

CLASSE →



Tipo: ?

Cor: ?

Placa: ?

Número de Portas: ?



Tipo: Porsche

Cor: Cinza

Placa: MHZ-4345

Número de Portas: 2

← OBJETOS →



Tipo: Ferrari

Cor: Vermelho

Placa: JKL-0001

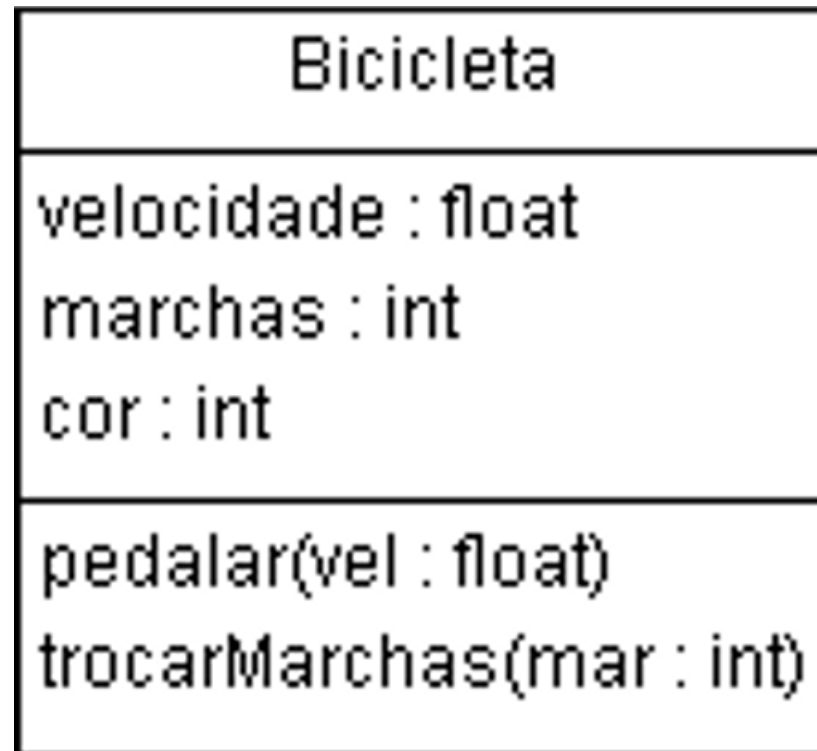
Número de Portas: 4

Exercícios

- ▶ Considerando o exemplo do slide anterior, para as classes Livro, Animal e Cachorro, descreva os atributos e comportamentos das classes e dos objetos correspondentes.

Veja que se aplicar os conceitos de banco de dados relacional, poderá ajudar na análise do exercício.

Representação de classe em UML



Descrição de uma classe em PHP

```
class Bicicleta {  
    // Propriedades (atributos) da bicicleta  
    public $marca;  
    public $modelo;  
    public $cor;  
    public $velocidadeAtual = 0;  
  
    // Método para iniciar a bicicleta com uma marca, mo  
    public function __construct($marca, $modelo, $cor) {  
        $this->marca = $marca;  
        $this->modelo = $modelo;  
        $this->cor = $cor;  
    }  
}
```

```
    // Método para acelerar a bicicleta  
    public function pedalarMaisRapido($incrementoVelocidade)  
    {  
        $this->velocidadeAtual += $incrementoVelocidade;  
        echo "Pedalando mais rápido! Velocidade atual: {$this->velocidadeAtual}";  
    }  
}
```

```
    // Método para frear a bicicleta  
    public function frear($decrementoVelocidade) {  
    }
```

Descrição de uma classe em PHP

```
// Método para exibir informações da bicicleta
public function mostrarInfo() {
    echo "Marca: {$this->marca}<br>";
    echo "Modelo: {$this->modelo}<br>";
    echo "Cor: {$this->cor}<br>";
    echo "Velocidade Atual: {$this->velocidadeAtual}";
}
}

// Criando uma instância da classe Bicicleta
$minhaBicicleta = new Bicicleta("Caloi", "Explorer", "Verde");

// Usando os métodos da bicicleta
$minhaBicicleta->mostrarInfo();
$minhaBicicleta->pedalarMaisRapido(10);
$minhaBicicleta->pedalarMaisRapido(15);
$minhaBicicleta->frear(5);
$minhaBicicleta->mostrarInfo();
```

10 Vantagens de programar OO

- ▶ A programação orientada a objetos (OO) oferece várias vantagens que contribuem para a organização, reutilização e manutenção de código. Aqui estão algumas das principais vantagens:
 - **1. Encapsulamento:** A OO permite encapsular os detalhes internos de uma classe, ocultando a complexidade e expondo apenas uma interface bem definida. Isso ajuda a proteger os dados internos de manipulações inadequadas e a facilitar a manutenção.
 - **2. Reutilização de Código:** Através do uso de herança e composição, é possível reutilizar código existente. Isso economiza tempo e esforço, pois você pode estender ou modificar classes existentes em vez de reescrever tudo do zero.

10 Vantagens de programar OO

- ▶ **3. Modularidade:** A OO incentiva a quebra do código em módulos (classes), o que torna o desenvolvimento mais organizado. Cada classe representa uma unidade funcional, facilitando a compreensão e o gerenciamento do código.
- ▶ **4. Flexibilidade e Extensibilidade:** Através do polimorfismo e da herança, você pode criar hierarquias de classes que podem ser estendidas e adaptadas para novos cenários sem afetar o código existente. Isso facilita a adição de novos recursos e a adaptação a mudanças nos requisitos.
- ▶ **5. Manutenção Simplificada:** Devido à modularidade e à estrutura hierárquica, a manutenção de código em um sistema orientado a objetos é mais direcionada e menos propensa a causar efeitos colaterais indesejados.

10 Vantagens de programar OO

- ▶ **6. Colaboração em Equipe:** A OO permite que diferentes desenvolvedores trabalhem em partes diferentes do sistema sem interferir nos outros. Cada classe define uma responsabilidade específica, o que facilita a divisão do trabalho em equipes maiores.
- ▶ **7. Modelagem do Mundo Real:** A OO permite modelar objetos do mundo real em código. Isso torna o desenvolvimento mais próximo da maneira como concebemos e entendemos os sistemas na vida real.
- ▶ **8. Polimorfismo:** Esse conceito permite tratar diferentes objetos de maneira uniforme, permitindo que você crie interfaces genéricas que funcionam para várias classes. Isso ajuda a reduzir a repetição de código e aumenta a flexibilidade.

10 Vantagens de programar OO

- ▶ **9. Abstração:** A OO permite criar classes abstratas que definem a estrutura e o comportamento geral, mas não podem ser instanciadas. Isso ajuda a fornecer uma base sólida para as classes derivadas.
- ▶ **10. Facilitação de Testes:** A OO facilita a criação de testes unitários, pois as unidades funcionais (classes) são bem definidas e isoladas, o que torna mais simples testar cada parte separadamente.
- ▶ Em resumo, a programação orientada a objetos promove uma abordagem mais organizada, reutilizável e adaptável ao desenvolvimento de software, tornando-o mais eficiente e manutenível ao longo do tempo.

Obrigado!!!

Profº Jânio Eduardo

janio.vasconcellos@gmail.com

(61) 98451-9188